



## 04 配列

作成者：GT F

最終更新日：2019-08-18

- [初期化](#)
- [要素のアクセス](#)
- [ArrayIndexOutOfBoundsException](#)
- [配列の各要素をアクセスする](#)
- [次元配列](#)
- [設問](#)

### 初期化

配列は、ある**特定の型**の要素**複数個**含めています。要素の個数は生成する時にサイズを指定します。下記例は、新しい配列生成構文を示します。

例1. 配列を初期化する（**型[]** 変数名）

```
1  int[] array1 = new int[4]; // {0, 0, 0, 0} int型の配列を初期化する（個数=4）
2  int[] array2 = new int[] {1, 2, 3, 4}; // int型の配列を初期化する（個数=4）
```

**i** 型 + [] => 配列型

```
1  int a = 10; // aはint型
2  System.out.println(a); // 10
3  int[] b = new int[10]; // bの型はintの配列型
4  System.out.println(b); // NG、特定の要素は？
```

### 要素のアクセス

配列の要素をアクセスする為、要素のindex（インデックス）を利用すること。使い方法は：配列の変数名[インデックス]

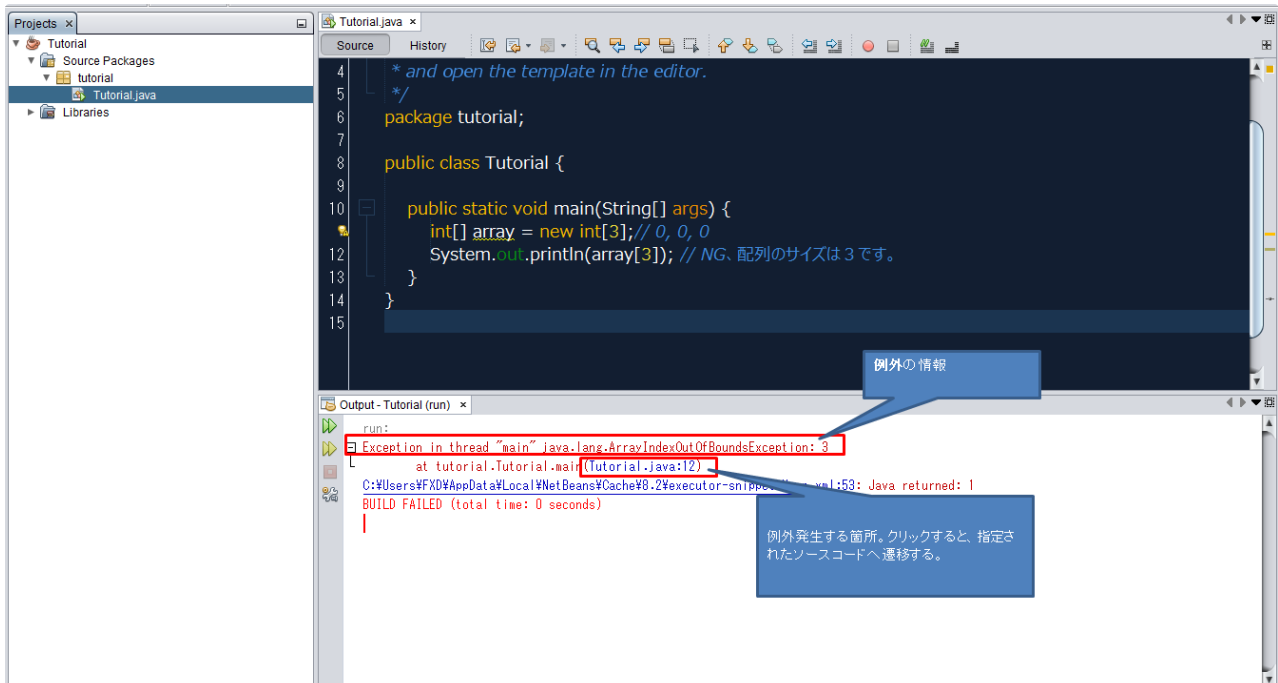
```
1  int[] array1 = new int[] {1, 2, 3, 4};
2  System.out.println(array1[0]); // 1
3  int[] array2 = new int[5];
4  System.out.println(array2[0]); // 0
```

**!** 配列の先頭要素のIndexは0です。

# ArrayIndexOutOfBoundsException

指定された配列のサイズを超える要素をアクセスするとき、例外が発生します。BUGの探す方法。

- 1 `int[] array = new int[3]; // 0, 0, 0`
- 2 `System.out.println(array[3]); // NG、配列のサイズは3です。`



## 配列の各要素をアクセスする

前提：`.`は日本語の「の」と同じ意味。変数の属性をアクセスする為。

- 1 `int[] array = new int[] {1, 2, 3, 0};`
- 2 `System.out.println(array.length); // array.Length = 4`

方法1：配列の各要素をアクセスする FOR

- 1 `int[] array = new int[] {5, 6, 7, 8};`
- 2 `for(int i = 0; i < array.length; i++) {`
- 3 `System.out.println(array[i]); // 配列の要素を出力する`
- 4 `}`

理解する手順：

1. 配列の第1位要素のアクセス方法 ⇒ `array[0]`
2. 配列の長さ ⇒ `array.length`
3. 配列最後の要素 ⇒ `array[ length - 1 ]`
4. 配列は各要素のインデックスは：`0, 1, 2, 3, ..., length - 1`
5. `for(int i = 0; i < array.length; i++) { }` `i`の値は `0, 1, 2, 3, ... length - 1`
6. `System.out.println(array[i]);` ⇒ 配列の各要素が可能。

## 方法 2 : 配列の各要素をアクセスする `foreach` → 拡張FOR文

```
1  int[] array = new int[] {5, 6, 7, 8};
2  for(int x : array) {
3      System.out.println(x); // 配列の要素を出力する
4  }
5  ↑ ↓ 同様
6  for(int i = 0; i < array.length; i++) {
7      int x = array[i];
8      System.out.println(x); // 配列の要素を出力する
9  }
```

⚠ 長さ(length)とインデックス(index)を迷わずに。長さは4の配列に、最後の要素のインデックスは 長さ-1 = 3であること。

## 次元配列

```
1  int[][] array2d = new int[][] {
2      {1, 2, 3, 4},
3      {21, 22, 23, 24}
4  };
5  System.out.println(array2d.length); // 出力 : 2
```

```
1  int[][] array2d = new int[4][];
2  array2d[0] = new int[] {1, 2, 3, 4, 5};
3  array2d[1] = new int[] {1, 2, 3, 4, 5};
4  array2d[2] = new int[] {1, 2, 3, 4, 5};
5  array2d[3] = new int[] {1, 2, 3, 4, 5};
```

```
1  int[][] array2d = new int[4][2];
2  array2d[0][0] = 1;
3  array2d[0][1] = 2;
4  // ...
```

## 設問

問題 1 : `int[] array = new int[4];` 各要素の値を記載してください。

問題 2 : `int[] array = new int[] {5, 6};` 各要素の値を記載してください。

問題 3 : `int[][] array = new int[2][3];` 各要素の値を記載してください。

問題 4 : 以下 2 重Foreachを理解してください。

```
1  int[][] array2d = new int[4][2];
```

```
2  for (int[] out : array2d) {  
3      for (int inner : out) {  
4          System.out.println(inner);  
5      }  
6  }
```

問題 5 : `int[] array = new int[] {1, 3, 5, 7, 100, 0, 1};` 左の配列の最大値と最小桁値を `System.out.println()`。

問題 6 : 問題 1 の配列のSUMを求める。

問題 7 : 行列（2次元配列）の和と積を計算してください。

 いいね 1 番に「いいね」しましょう

ラベルがありません 